



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/816,493	04/01/2004	Muralidharan Lakshminarasimhan Kanchi	IPP03-01	8597
24628	7590	12/15/2008		
Husch Blackwell Sanders, LLP			EXAMINER	
Husch Blackwell Sanders LLP Welsh & Katz			WON, MICHAEL YOUNG	
120 S RIVERSIDE PLAZA				
22ND FLOOR			ART UNIT	
CHICAGO, IL 60606			PAPER NUMBER	
			2455	
			MAIL DATE	
			DELIVERY MODE	
			12/15/2008	
			PAPER	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/816,493

**Applicant(s)**KANCHI, MURALIDHARAN  
LAKSHMINARASIMHAN**Examiner**

MICHAEL Y. WON

**Art Unit**

2455

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 24 October 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| <p>1) <input type="checkbox"/> Notice of References Cited (PTO-892)</p> <p>2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)</p> <p>3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br/>Paper No(s)/Mail Date _____</p> | <p>4) <input type="checkbox"/> Interview Summary (PTO-413)<br/>Paper No(s)/Mail Date _____</p> <p>5) <input type="checkbox"/> Notice of Informal Patent Application</p> <p>6) <input type="checkbox"/> Other: _____</p> |
|--|---|

### DETAILED ACTION

1. This action is in response to the amendment filed October 24, 2008.
2. Claims 1, 9, 17, and 22 have been amended.
3. Claims 1-30 have been examined and are pending with this action.

### ***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-30 are rejected under 35 U.S.C. 102(b) as being anticipated by Eichstadt et al. (US 2003/0023754).

### **INDEPENDENT:**

As per **claim 1**, Eichstadt teaches a method for executing a requested component of a script, comprising:

transmitting parameter information about a requested component from a server to a client (see page 1, [0008]: "transmits the modified web-page including the inserted script code to the one or more users who requested the web-page"), wherein the parameter information comprises at least one of specific information about user

interfaces (see page 6, [0042]: "The script code 400 may provide a toolbar 420, 420' that may contain..."), set of field names and types, and linking and interrelationship information;

linking the transmitted component parameter information to a predefined structure at the client to provide a script specific predefined structure, the predefined structure having an intended functionality corresponding to the intended functionality of the requested component (see page 1, [0010]: "That script code comprises the client component of the software and will add functionality to the web-page"); and,

executing the script specific predefined structure to execute the component (see page 4, [0030]: "the initialization script originally provided with the web-page is executed, followed by the execution of the inventive script code").

As per **claim 9**, Eichstadt teaches a system for executing a component of a script, comprising:

a client including a client memory, a client processor, and a client transceiver in communication with one another (see page 3, [0023]), the client memory including component script transmitted by a server (see page 1, [0008]: "transmits the modified web-page including the inserted script code to the one or more users who requested the web-page"; and [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM"), the transmitted component script including parameter information about the component, wherein the parameter information comprises at least one of specific information about user interfaces (see page 6, [0042]: "The script code 400 may provide a toolbar 420, 420' that may contain..."), set of field names and types,

and linking and interrelationship information and the client memory further including a client predefined structure having an intended functionality corresponding to an intended functionality of the component, wherein the processor is configured to link the parameter information of the transmitted component script to the client predefined structure to provide a script specific predefined structure (see page 1, [0010]: "That script code comprises the client component of the software and will add functionality to the web-page") and to execute the component by executing the script specific predefined structure (see page 4, [0030]: "the initialization script originally provided with the web-page is executed, followed by the execution of the inventive script code").

As per **claim 17**, Eichstadt teaches an application for executing a component of a script when a user runs the component on a system, the application comprising:

a first run time engine comprising an execution engine comprising a predefined structure and a linker, the predefined structure having an intended functionality of one of a plurality of component types, wherein the component has the intended functionality of one of the plurality of component types (see page 3, [0025]: "a user may enter that predetermined Internet address in their browser and cause their computer to..."), and wherein, when the user runs the component:

(a) the linker instructs a client processor to link parameter information about the component to the predefined structure to provide a script specific predefined structure, the parameter information being transmitted from a server to a client and stored in a client processor readable memory (see page 1, [0008]: "transmits the modified web-page including the inserted script code to the one or more users who requested the

web-page"; and [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM"), wherein the parameter information comprises at least one of specific information about user interfaces (see page 6, [0042]: "The script code 400 may provide a toolbar 420, 420' that may contain..."), set of field names and types, and linking and interrelationship information, and

(b) the execution engine instructs the client processor to execute the script specific predefined structure to execute the component (see page 4, [0030]: "the initialization script originally provided with the web-page is executed, followed by the execution of the inventive script code");

wherein the first run time engine is stored in a media and the first run time engine is transferred to a client processor readable memory of a system including the client processor readable memory and the client processor when the media is used with the system (see page 1, [0008]: "transmits the modified web-page including the inserted script code to the one or more users who requested the web-page"; and [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM").

As per **claim 22**, Eichstadt teaches a system for executing a component of a script, comprising:

a memory including component script, the component script including parameter information about the component, wherein the parameter information comprises at least one of specific information about user interfaces (see page 6, [0042]: "The script code 400 may provide a toolbar 420, 420' that may contain..."), set of field names and types, and linking and interrelationship information, and the memory further including a

predefined structure having an intended functionality corresponding to an intended functionality of the component (see page 3, [0025]: "a user may enter that predetermined Internet address in their browser and cause their computer to..."); and,

a processor in communication with the memory, wherein the processor is configured to link the parameter information of the component script to the predefined structure to provide a script specific predefined structure (see page 1, [0010]: "That script code comprises the client component of the software and will add functionality to the web-page"),

to temporarily store the script specific predefined structure in the memory (see page 1, [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM"),

to execute the component by executing the script specific predefined structure (see page 4, [0030]: "the initialization script originally provided with the web-page is executed, followed by the execution of the inventive script code"), and

to automatically delete the script specific predefined structure when the user exits the component (see pages 1-2, [0010]: "the script code is erased from RAM and thus no longer available to the user's browser").

As per **claim 26**, Eichstadt teaches a method for executing a component of a script of a program, the method comprising:

transferring a predefined structure to a memory, the predefined structure having an intended functionality corresponding to an intended functionality of one of a plurality

of component types (see page 3, [0025]: "a user may enter that predetermined Internet address in their browser and cause their computer to..."),

linking component script to the predefined structure to provide a script specific predefined structure, wherein the component has the intended functionality of one of the plurality of component types (see page 1, [0010]: "That script code comprises the client component of the software and will add functionality to the web-page"),

temporarily storing the script specific predefined structure in the memory (see page 1, [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM");

executing the component by executing the script specific predefined structure (see page 4, [0030]: "the initialization script originally provided with the web-page is executed, followed by the execution of the inventive script code"); and,

automatically deleting the script specific predefined structure from the memory after the program has been exited (see pages 1-2, [0010]: "the script code is erased from RAM and thus no longer available to the user's browser").

As per **claim 29**, Eichstadt teaches an application for executing a component of a script when a user runs the component on a system, the application comprising:

a first run time engine comprising an execution engine comprising a predefined structure and a linker, the predefined structure having an intended functionality of one of a plurality of component types, wherein the component has the intended functionality of one of the plurality of component types (see page 3, [0025]: "a user may enter that



predetermined Internet address in their browser and cause their computer to..."), and wherein, when the user runs the component:

(a) the linker instructs a processor to link parameter information about the component to the predefined structure to provide a script specific predefined structure (see page 1, [0010]: "That script code comprises the client component of the software and will add functionality to the web-page"),

(b) the execution engine instructs the processor to temporarily store the script specific predefined structure in a processor readable memory (see page 1, [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM"), and

(c) the execution engine instructs the processor to execute the script specific predefined structure to execute the component (see page 4, [0030]: "the initialization script originally provided with the web-page is executed, followed by the execution of the inventive script code"), and, when the user exits the application, the execution engine instructs the processor to automatically delete the script specific predefined structure from the processor readable memory (see pages 1-2, [0010]: "the script code is erased from RAM and thus no longer available to the user's browser");

wherein the first run time engine is stored in a media and the first run time engine is transferred to a processor readable memory of a system including the processor readable memory and the processor when the media is used with the system (see page 1, [0008]: "transmits the modified web-page including the inserted script code to the one or more users who requested the web-page"; and [0010]: "the script code is not permanently stored on the client computer, but loaded into RAM").

DEPENDENT:

As per **claim 2**, which depends on claim 1, Eichstadt teaches further comprising sending a request for the component from the client to the server (see page 1, [0010]).

As per **claim 3**, which depends on claim 2, Eichstadt teaches further comprising searching for the requested component in the script at the server (see page 1, [0010]).

As per **claim 4**, which depends on claim 1, Eichstadt further teaches wherein the linking step further comprises locating identifiers within the parameter information and inserting script data associated with the identifiers into the predefined structure based on corresponding identifiers in the predefined structure (see page 2, [0013]).

As per **claim 5**, which depends on claim 1, Eichstadt teaches further comprising determining the access level of the user, wherein the transmitting step further comprises transmitting the parameter information based on the user access level(see page 3, [0028]).

As per **claim 6**, which depends on claim 1, Eichstadt teaches further comprising storing the predefined structure at the client and storing a copy of the predefined structure at the server so that there is a client predefined structure, and a server predefined structure (see page 1, [0008] & [0009]).

As per **claim 7**, which depends on claim 1, Eichstadt teaches further comprising automatically deleting the script specific predefined structure after a user has exited the component (see pages 1-2, [0010]).

As per **claim 8**, which depends on claim 1, Eichstadt teaches further comprising the client sending a request for the component to the server to establish a connection (see page 3, [0025]), and the server creating a session identification number for the connection so that the client and the server can follow a connectionless protocol (inherency).

As per **claim 10**, which depends on claim 9, Eichstadt teaches further comprising a server in communication with the client, the server including a server memory, a server processor, and a server transceiver in communication with one another, the server memory including the script, the server transceiver being configured to transmit the component script, wherein the component script includes the parameter information about the component (see page 1, [0010] and page 3, [0023]).

As per **claim 11**, which depends on claim 9, Eichstadt further teaches wherein the server memory further comprises a server predefined structure having an intended functionality corresponding to an intended functionality of one of a plurality of component types, wherein the component has the intended functionality of one of the plurality of component types (see page 1, [0008] and page 3, [0026]).

As per **claim 12**, which depends on claim 9, Eichstadt further teaches wherein the client further comprises a client run time engine stored in the client memory, the client run time engine including a client parser and a client execution engine, the client execution engine including a client linker and the client predefined structure, the client parser configured to instruct the processor to search for identifiers in the transmitted component script, the client linker configured to instruct the client processor to link the

parameter information to the client predefined structure to provide the script specific predefined structure (see page 4, [0031]-[0032]).

As per **claim 13**, which depends on claim 12, Eichstadt teaches further comprising a server in communication with the client, the server including a server memory, a server processor, and a server transceiver in communication with one another, the server memory including the script and a server run time engine, the server run time engine including a server parser and a server execution engine, the server execution engine including a server linker and a server predefined structure having an intended functionality corresponding to an intended functionality of one of a plurality of component types, wherein the component has the intended functionality of one of the plurality of component types, the server parser configured to instruct the server processor to search for the component in the script, the component being requested by the client and comprising the component script including the parameter information about the component, the server linker configured to instruct the server processor to link the parameter information to the server predefined structure to provide a server script specific predefined structure; and, the server transceiver being configured to transmit the component script, wherein the server and the client have the same intelligence with respect to the client and server run time engines (see page 3, [0026]-[0030]).

As per **claim 14**, which depends on claim 9, Eichstadt further teaches wherein the component script is transmitted from the server to the client when the client requests the component script and wherein the client memory further comprises a client long term memory and a client short term memory, the run time engine being stored in the

client long term memory before the client requests the component script, wherein the client processor is configured to transfer the run time engine to the client short term memory when the client requests the component script, to temporarily store the script specific predefined structure in the client short term memory, and to automatically delete the script specific predefined structure from the client short term memory when the client exits the component (see pages 1-2, [0010] and page 3, [0023]).

As per **claim 15**, which depends on claim 9, Eichstadt further teaches wherein the transmitted parameter information includes identifiers associated with component information and the predefined structure includes corresponding identifiers (see page 5, [0037]).

As per **claim 16**, which depends on claim 9, Eichstadt further teaches wherein the server creates a unique session identification number for every connection established to uniquely identify each connection and recreate the session previously established thereby facilitating a connectionless protocol (inherency).

As per **claim 18**, which depends on claim 17, Eichstadt further teaches wherein a server run time engine is transferred to a server processor readable memory of the system and the server run time engine comprises a copy of the first run time engine, wherein the system includes a server comprising a server processor readable memory, a server transceiver, and a server processor, wherein the server is in communication with a client, the client comprising the client processor and the client processor readable memory; the server run time engine comprising a server parser and a server execution engine, wherein a user at the client requests a component from the server prior to

running the component and, when the user requests the component: (a) the server parser instructs the server processor to search the script for the requested component, the script being stored in the server processor readable memory, and (b) the execution engine instructs the server processor to transmit component script including the parameter information about the component to the client via the server transceiver (see page 3, [0026]-[0030]).

As per **claim 19**, which depends on claim 18, Eichstadt further teaches wherein the server execution engine further comprises a server predefined structure, the server predefined structure having the intended functionality of one of the plurality of component types, wherein the requested component has the intended functionality of the one of the plurality of component types (see page 3, [0026]-[0030]).

As per **claim 20**, which depends on claim 18, Eichstadt further teaches wherein, when the client requests the component (see page 3, [0025]), the server execution engine instructs the server processor to create a session number and to transmit the session number to the client (see page 3, [0028]).

As per **claim 21**, which depends on claim 17, Eichstadt further teaches wherein the execution engine instructs the client processor to store the script specific predefined structure in the client processor readable memory and instructs the processor to automatically delete the script specific predefined structure from the memory after the user exits the component (see pages 1-2, [0010]).

As per **claim 23**, which depends on claim 22, Eichstadt further teaches wherein the script further comprises script transmitted from a server (see pages 1-2, [0010]).

As per **claim 24**, which depends on claim 23, Eichstadt teaches further comprising a server in communication with the client, the server including a server memory, a server processor, and a server transceiver in communication with one another, the server memory including the script, and the server transceiver being configured to transmit the component script (see page 1, [0010] and page 3, [0023]).

As per **claim 25**, which depends on claim 24, Eichstadt further teaches wherein the server memory further comprises a server predefined structure having an intended functionality corresponding to an intended functionality of one of a plurality of component types, wherein the component has the intended functionality of one of the plurality of component types (see page 1, [0008] and page 3, [0026]).

As per **claim 27**, which depends on claim 26, Eichstadt teaches further comprising storing the predefined structure in a long term memory, wherein the transferring step further comprises transferring the predefined structure to a short term memory when a user requests a component, the storing step further comprises storing the script specific predefined structure in the short term memory, and the automatically deleting step further comprises automatically deleting the script specific predefined structure from the short term memory (see pages 1-2, [0010] and page 3, [0023]).

As per **claim 28**, which depends on claim 26, Eichstadt teaches further comprising transmitting component script to a client from a server, wherein the transferring step further comprises transferring the predefined structure to a client memory, the storing step further comprises storing the script specific predefined structure in the client memory, and the automatically deleting step further comprises

automatically deleting the script specific predefined structure from the client memory (see pages 1-2, [0010]).

As per **claim 30**, which depends on claim 29, Eichstadt further teaches wherein a server run time engine is transferred to a server processor readable memory of the system and the server run time engine comprises a copy of the first run time engine, wherein the system includes a server comprising a server processor readable memory, a server transceiver, and a server processor, wherein the server is in communication with a client, the client comprising the processor and the processor readable memory; the server run time engine comprising a server parser and a server execution engine, wherein a user requests a component prior to running the component and when the user requests the component: (a) the server parser instructs the server processor to search the script for the requested component, the script being stored in the server processor readable memory, and (b) the execution engine instructs the server processor to transmit parameter information of the component to the client via the server transceiver (see page 3, [0026]-[0030]).

### ***Response to Arguments***

5. Applicant's arguments filed May 19, 2008 have been fully considered but they are not persuasive. Eichstadt clearly and explicitly teach all the limitations of the recited claims.

In particular, Eichstadt clearly teaches linking the toolbar 420, 420' (script) to the browser interface to render the interactive web page (see page 6, [0042]-[0043]).



Clearly the combination of Eichstadt teachings teach "linking the transmitted component parameter information to a predefined structure at the client to provide a script specific predefined structure, the predefined structure having an intended functionality corresponding to the intended functionality of the requested component".

### ***Conclusion***

6. For the reasons above, claims 1-30 have been rejected and remain pending.
7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL Y. WON whose telephone number is (571)272-3993. The examiner can normally be reached on M-Th: 10AM-8PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Michael Won/

Primary Examiner

December 8, 2008

**Application Number****Application/Control No.**

10/816,493

**Applicant(s)/Patent under  
Reexamination**KANCHI, MURALIDHARAN  
LAKSHMINARASIMHAN**Examiner**

MICHAEL Y. WON

**Art Unit**

2455